CSE 4502/5717 Big Data Analytics
Spring 2026 Exam 3 Helpsheet

1. **Association Rules Mining.** An **itemset** is a set of items. A k**-itemset** is an itemset of size k. A **transaction** is an itemset. A **rule** is represented as $X \to Y$ where $X \neq \emptyset, Y \neq \emptyset, X \cap Y = \emptyset$.

   We are given a database DB of transactions and the number of transactions in the database is n. Let I be the set of distinct items in the database and let $d = |I|$.

   For an itemset X, we define $\sigma(X)$ as the number of transactions in which X occurs, i.e. $\sigma(X) = |\{T \in DB | X \subseteq T\}|$ The **support** of any rule $X \to Y$ is $\frac{\sigma(X \cup Y)}{n}$. The **confidence** of any rule $X \to Y$ is $\frac{\sigma(X \cup Y)}{\sigma(X)}$.

   Association Rules Mining is defined as follows.

   Input: A DB of transactions and two numbers: minSupport and minConfidence.
   Output: All rules $X \to Y$ whose support is $\geq$ minSupport and whose confidence is $\geq$ minConfidence.

   An itemset is **frequent** if $\sigma(X) \geq n \cdot \text{minSupport}$

   We discussed the Apriori algorithm for finding all the frequent itemsets. This algorithm is based on the a priori principle: If X is not frequent then no superset of X is frequent. Also, If X is frequent then every subset of X is also frequent.

   The pseudocode for the Apriori algorithm is given next.

Algorithm 1: Apriori algorithm

```
k := 1;
Compute F_1 = {i ∈ I | σ(i) ≥ n · minSupport};
while F_k ≠ ∅ do
    k := k + 1;
    Generate candidates C_k from F_{k−1};
    for T ∈ DB do
        for C ∈ C_k do
            if C ⊆ T then
                σ(C) := σ(C) + 1;
    F_k := ∅;
    for C ∈ C_k do
        if σ(C) ≥ n · minSupport then
            F_k := F_k ∪ {C};
```

   We can use a hash tree to compute the support for each candidate itemset.

   We also presented a randomized Monte Carlo algorithm for identifying frequent itemsets. The idea was to pick a random sample, identify frequent itemsets in the sample (with a smaller support) and output these. We proved that the output of this algorithm will be correct with a high probability using the Chernoff bounds:

   If X is $B(n, p)$, then the following are true:

$$Prob.[X \geq (1+\epsilon)np] \leq \exp(-\epsilon^2 np/3)$$

$$Prob.[X \leq (1-\epsilon)np] \leq \exp(-\epsilon^2 np/2),$$

for any $0 < \epsilon < 1$.

2. **Hierarchical Clustering.** We showed that we can perform hierarchical clustering on n given points in $O(n^2)$ time. The idea of hierarchical clustering is to start with n clusters where each input point is a cluster on its own. From thereon, we merge the two closest clusters at a time until a termination condition is reached.

3. **Quantum Computing.** We introduced quantum circuits. We discussed the ideas of entanglement and teleportation. We also stated the no cloning theorem. In addition, we also gave the details of Grover's algorithm. Grover's algorithm solves the following problem: Given a sequence $X = x_1, x_2, \ldots, x_N$ and a function $f : X \to \{0, 1\}$, find an i such that $f(x_i) = 1$. Grover's algorithm solves this problem in $O(\sqrt{N})$ time using a quantum circuit. The output of this circuit will be correct with a constant probability.